Procesamiento del Lenguaje Natural, num. 29 (2002), pp.105-111 recibido 27-05-2002; aceptado 01-07-2002

Morphological Analysis of Inflective Languages through Generation*

Alexander Gelbukh and **Grigori Sidorov**
Natural Language Laboratory,
Center for Computing Research (CIC),
National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, CP 07738, Zacatenco, México DF
{gelbukh, sidorov}@cic.ipn.mx

Abstract: A crucial problem in development of systems for automatic morphological analysis for inflective languages is the treatment of stem alternations. The existing models require development of the corresponding rules that specify what stems can be generated from a given one. Many of such rules (e.g., for Russian about a thousand) do not have any reasonable linguistic interpretation. We suggest a method that avoids the use of such rules by generating and verifying the hypotheses about possible grammatical forms. The methods of such type are known as analysis through generation; they make the system development much simpler than the standard direct approach. A morphological analysis and generation system for Russian developed with our method is freely available for academic use; a Spanish system is being implemented.
Keywords: automatic morphological analysis, inflective languages, analysis through generation.

Resumen: Un problema crucial en el desarrollo de los sistemas para el análisis morfológico automático de los idiomas flexivos es el tratamiento de las alternaciones de la base. Los modelos existentes requieren el desarrollo de las reglas correspondientes que especifican qué variantes de la base se pueden generar de la variante dada. Un gran número de tales reglas (por ejemplo, para el lenguaje ruso alrededor de un mil) no tiene ninguna interpretación lingüística razonable. Sugerimos un método que evite el uso de tales reglas gracias a la generación y verificación de las hipótesis sobre las formas gramaticales posibles. Los métodos de este tipo –conocidos como análisis a través de generación– hacen el desarrollo de sistemas mucho más simple que el el enfoque directo estándar. Un sistema para el análisis y la generación morfológica para el lenguaje ruso, desarrollado con nuestro método está disponible sin costo para el uso académico; el sistema para el español está bajo desarrollo.
Palabras clave: análisis morfológico automático, lenguajes flexivos, análisis a través de generación.

1 *Introduction*

The methods for automatic morphological analysis can be classified into dictionary-based and heuristic-based ones. The former ones use a stem dictionary to guarantee the correct results for the words stored in the dictionary. The latter ones use heuristic rules to guess the result for previously unseen words, which is important since new words constantly appear in the language, not mentioning that no dictionary can be complete.

In this paper, we are mostly concerned with the former type of models (though will touch upon the latter type, too, see Section 8 below).

One of the most famous models for morphological analysis is the two-level model (KIMMO) suggested by Koskenniemi (1983). There exist a number of other models for different languages (Gelbukh 2002, Hausser 1999,

* Work done under partial support of Mexican Government (CONACyT and SNI) and CGEPI-IPN, Mexico.

Sedlacek and Smrz 2001, Sidorov, 1996, Yablonsky, 1999).

The reason for this diversity is that different languages have different morphological structure; the methods perfectly suitable for morphologically poor languages (like English) or agglutinative languages (like Finnish) are not the best ones for inflective languages (like Spanish or Russian).

In theory, since the morphological system of any inflective language is finite, any dictionary-based method of analysis gives equally correct results. However, not all methods are equally convenient to use and easy to implement.

At one extreme is storing all grammatical forms in a dictionary, along with the lemma and all necessary grammatical information associated with the form. With this approach, a morphological system is just a very large two-column database. This is possible for inflective languages (though not for agglutinative or polysynthetic ones). Modern computers have the possibility of storing databases containing all grammatical forms for large dictionaries of inflective languages (a rough approximation for Spanish and Russian is 20 to 50 megabytes).[1]

Yet applications that use algorithms to reduce the dictionary size to, say, 1 megabyte, are preferable. Indeed, a morphological analyzer is usually used together with a syntactic parser, semantic analyzer, and a reasoning or retrieval engine, so that freeing physical memory for these modules is highly desirable. Note that the use of large virtual memory makes simultaneous access to very large data structures possible, but does not make it faster since the data are anyway stored physically on the hard disk.

Such algorithmic solutions have a number of additional advantages. For example, an analysis algorithm can include heuristics for recognition of unknown (new) words using the dictionary as a knowledge base for its heuristics.

In this paper we suppose that the analysis algorithm works, in outline, in the following classical way (sometimes called stripping method). The input wordforms are analyzed one by one. From each wordform, a number of substrings from some fixed lists (suffixes, flexions, particles, etc.) are detached. What remains is expected to be the stem and is looked up in the stem dictionary. The analysis is considered successful if this substring is really found in the dictionary and the grammatical information it is supplied with in this dictionary indicates that this stem is compatible with the set of affixes previously detached (Hutchins, Somers, 1992).

A crucial issue in the development of an algorithmic analysis system is the treatment of regular stem alternations (English *stop-∅ – stopp-ing*, Spanish *pens-ar – p<u>ie</u>ns-a* ‘to think–thinks , Russian *молото<u>к</u>-∅ – молотк-а* ‘hummer–of hummer). Explicit specification in the dictionary of all such variants, together with the associated grammatical information, is boring and leads to redundancy.

On the other hand, in the existing systems the rules used for automatic recognition of non-first stems (usually by guessing the first stem: given *молотк-*, guess *молото<u>к</u>-*) are numerous, complicated, and anti-intuitive. Our point here is that the inverse operation – given *молото<u>к</u>-*, guess *молотк-* – in many cases is much simpler, and the corresponding rules are well-known, easily expressed, and easily programmable.

In this paper we discuss how to develop a morphological analysis system for an inflective language with less effort and applying more intuitive and flexible morphological models. We show that the use of a non-straightforward method can greatly simplify the analysis procedure and allows using morphological models much more similar to the traditional grammars.

We avoid development of stem transformation rules oriented to analysis and to use the generation module instead (this idea is known as analysis through generation). Our implementation, however, will require storing in the morphological dictionary all stems for each word with the corresponding information[2].

In the rest of the paper, we first describe the suggested method in detail. First we describe the types of morphological information we use. Then we discuss the morphological models

[1] 100 thousand lexemes give approximately 1 million grammatical forms (in Russian, 8.2 forms per lexeme on average was reported), each one being 10-byte long, plus some 10 bytes of the lemma, plus some grammatical information.

[2] In our implementation, this information is stored in the form of the lexeme identifier (a number) and a small pointer to the grammeme string shared by many dictionary entries.

Any information (syntactic, semantic, etc.) associated with the lexeme rather than with a specific grammatical form and irrelevant for the process of morphological analysis *per se* is stored separately along with the lexeme identifier and is returned among the analysis results.

(and the corresponding algorithms) we have used to implement the method. Then we describe the functioning of our method: analysis, generation, and the treatment of unknown (new) words. Finally, we briefly discuss the implementation for Russian and Spanish languages.

2 The Method

As we have mentioned, a major problem of automatic morphological analysis of inflective languages is stem alternations. E.g., the forms *stop* and *stopp-ed* use two different stems. The direct way to handle such alternations is constructing the rules that take into account all possible stem alternations during the analysis process; for example, for Russian the number of such rules is about a thousand (Malkovsky, 1985).

However, such rules do not have any correspondence in traditional grammars, i.e., they have no intuitive correspondence in language knowledge. In addition, too many such rules are necessary.

Another possibility is to store all stems in the dictionary, together with the information on their possible grammatical categories; this method has been used for Russian (Gel bukh 1992) and Czech (Sedlacek and Smrz 2001). We adopt this possibility, but propose a different technique for treatment of grammatical information: our technique is dynamic while the techniques described in those papers are static.

We apply the technique known as analysis through generation. Since analysis is usually more complex than generation, this technique allows for simpler implementation.

3 Types of Grammatical Information

We use two knowledge sources:

- The stem dictionary and
- A list of grammatical categories for each part of speech.

The stem dictionary independently stores all variants of stems for each lexeme. For example, in Spanish verbs with alternations usually have two or three stems (except, e.g., *ment-ir*, *mient-o*, *mint-ió*, *miént-a-le* ‘to lie, I lie, he lied, lie to him!)and some nouns and adjectives have two stems (e.g., *francés – francesa*, *carácter – caracteres*, *régimen – regímenes*); in Russian, nouns with alternations have two stems and verbs up to four stems. A separate dictionary entry corresponds to every such stem. Together with the stem, the entry contains the information necessary for word form generation, such as:

- The stem number (first stem, second stem, etc.).
- Part of speech.
- The presence of alternations.
- Morphological type. For example, for Spanish nouns: gender, for Spanish verbs: stem alternation class, for Russian nouns: word formation type for each of the three genders — say, for feminine there are 7 types, etc.
- Additional marks. For example, the absence of the singular form (pluralia tantum), like in Spanish *anteojos* ‘spectacles ; the presence the prepositional case variation for Russian nouns like *в шкафу* ‘in wardrobe versus *о шкафе* ‘about wardrobe , etc.

The list of grammatical categories stores for a given part of speech[3] all possible categories represented as sets of grammemes such as “singular” (a value of the category “number”) or “nominative case” (a value of the category “case”). Any grammatical form is characterized by a combination of grammemes. For example, for Russian nouns the list consists of the case and number; for Russian full adjectives: case, number, and gender; for Spanish nouns: number (singular or plural), etc. An example of a Spanish verbal grammatical category[4] is “indefinite preterit, indicative, singular, second person.”

4 Types of Morphological Models

Three morphological models are used:

- The correspondence between the flexions and the grammemes,
- The correspondence between the stems (stem numbers) and the grammemes,
- The correspondence between alternating stems of the same lexeme.

The first model establishes the correspondence between the flexions and grammatical categories (sets of grammemes), taking into account different grammatical types fixed in the dictionary, e.g., Spanish *-aba* or *-ía* ⇔ “imperfect preterit, indicative, singular, first or third

[3] For this reason, we do not call the part of speech a grammatical category, which is a pure matter of terminology.

[4] Again, different terminology is used in the literature.

person" as in *hablaba* 'was speaking or *comía* 'was eating .

In the process of analysis, we use the correspondence "flexions ⇒ sets of grammemes" (to formulate hypothesis), and in the process of generation, the correspondence "sets of grammemes ⇒ flexions."

A similar correspondence is established between the sets of grammemes and the types (numbers) of stems; however, this correspondence is used only for generation. For example, if a Russian feminine noun of a certain type has a stem alternation, then the first stem is used for all forms except for genitive case plural, for which the second stem is used (e.g.: *сосн-а – сосен-∅* 'pine – of pines); for a Spanish verbs of the type *pensar* 'think the second stem *piens-* is used for the forms of present indicative or subjunctive singular all persons and plural third person, while for the other forms the first stem *pens-* is used.

Note that we do not need to formulate the corresponding model for analysis, which makes our method simpler than direct analysis.

To be able to generate all forms starting from a given one it is necessary to be able to obtain all stems of the lexeme in question from the given one. There are two ways to do this: static and dynamic, which have their own pros and contras. The static method implies storing in the dictionary together with each stem the correspondence between the stems (e.g., each stem has a unique identifier by which different stems of a lexeme are linked in the dictionary).

Storing the explicit links increases the size of the dictionary. Thus, we do this dynamically. It was sufficient to develop the algorithm for constructing (1) the first stem (that of the dictionary form, e.g., infinitive) from any other stem and (2) any other stem from this first stem. In this way, starting from any stem we can generate any other stem. To construct all stems from the first stem in runtime, we used the algorithm that had been implemented anyway for dictionary compilation.

The difference between static and dynamic methods is that in the former case the stem generation algorithm is applied in the compile time (when the dictionary is built) while in the latter case in runtime, which does not affect performance significantly and can even slightly speed up the processing because the smaller dictionary is better cached in memory.

Note that the rules of these algorithms are different from those used for direct analysis. For Russian, we use about 50 stem-construction rules, which do not significantly differ from those taught to foreigners learning Russian. For example, the rule:

$$\text{-VC \& A1} \Rightarrow \text{-C} \qquad (1)$$

means: if the stem ends in a vowel (V) following by a consonant (C) and the stem alternation of type 1 (A1) is present then the vowel is removed. Applied to the first stem of the Russian noun *молоток* 'hammer ,the rule generates the stem *молотк-* of the word form *молотка* 'of hammer .

In Spanish, there are few alternations and thus few such rules. For example, for the verb *conocer* 'to know as well as other verbs of the alternation type 13, the first stem is *conoc* and the rule for generation of the second stem *conozc* is:

$$\text{-C \& A13} \Rightarrow \text{-}z\text{C} \qquad (2)$$

The small amount of our simple generation rules contrasts with about 1000 very superficial and anti-intuitive rules necessary for direct analysis. For example, to analyze a non-first-stem word in Russian, Malkovsky (1985) uses the rules that try to invert the effect of (1): if the stem ends in a consonant, try to insert a vowel before it and look up each resulting hypothetical stem in the dictionary: for *молотк-*(*а*), try *молотк-*, *молотек-*, *молоток-*, etc. This also affects the system performance.

Two considerations explain the simplicity of our rules. First, we use the information about the alternation type of the stem, stored in the dictionary. For Russian, this information can be borrowed from the dictionary by Zalizniak (1980); for Spanish, which has more regular morphology, the list of words with stem alternations is given in any large bilingual dictionary.

Second, often generation of a non-first stem from the first one is simpler than vice versa. More precisely, the stem that appears in the dictionaries for a given language is the one that allows simpler generation of other stems (note that in some languages the dictionary form for verbs is not the infinitive: say, in Hebrew this is third person past singular masculine).

5 Data Preparation

We needed some preliminary data preparation work that consisted of the following main steps:

- Describing and classifying all words of the given language into grammatical classes (usually this information can be found in the existing dictionaries);
- Converting the available lexical information into a stem dictionary (only the first stem needs to be generated at this step);
- Applying the algorithms of stem generation (first stem ⇒ other stems) to generate all stems;
- Generating the stem numbers for each (non-first) stem.

To perform the last two steps, the data record generated for the first stem is copied, the stem is changed to obtain the required form, and the stem number mark is updated.

6 Generation Process

Given a word form of a lexeme and the required grammatical category (set of grammemes), the corresponding word form is to be constructed. E.g., it is required to construct the imperfect preterit second person singular of *piensa* ‘thinks .

For this, the following steps are executed:

- The model “grammatical category ⇒ stem number” is applied to find the necessary stem number,
- The necessary stem is generated,
- The corresponding data from the stem dictionary is retrieved,
- Using this information, the correct flexion is chosen and concatenated with the stem.

To generate a non-first stem is to be used then we generate the first stem and from it, the necessary stem.

If necessary, this process is repeated to add more than one flexion to the stem. For example, Russian participles (verbal forms) use the same flexions as adjectives to express the number, case, and gender and also special suffixes to indicate that this is a participle, i.e., they are concatenations of a stem and two affixes (*пиш-ущ-ий* ‘which masculine singular is writing). In this case, we first generate the participle stem *пиш-ущ-* by adding the suffix (using the dictionary information on the properties of the corresponding verbal stem) and then use the information for an adjective of the corresponding declension type to add the flexion *-ий*.

Both in Russian and Spanish such repetition is limited to only three steps, the “longest” forms being, e.g., Russian *пиш-ущ-ий-ся* ‘which masculine singular is written and Spanish *dá-ndo-me-lo* ‘giving it to me .

In some cases, such splitting is ambiguous, e.g. Spanish *como* ‘as vs. *com-o* ‘I eat . In such cases a recursive algorithm is used to find all possible combinations of a valid stem and a valid set of affixes. Since such cases are rather rare and the number of possible combinations is small, this does not present in practice any computational complexity problems. Probably this is due to the fact that our language is optimized to avoid difficult garden path constructions.

Finally, note that since we use precise information on the set of forms allowed for a specific stem and the affixes used for their formation, our algorithm does not present any over- or undergeneration problems – of course, at the cost of a large dictionary and impossibility to process unknown words (cf. Section 8).

7 Analysis Process

Given a letter string (a word form) in the input, we analyze it in the following way:

1. The letters are separated one by one from right to left to get the possible flexion: given *stopping*, we try *-∅* (zero flexion), then *-g*, *-ng*, *-ing*, *-ping*, etc.; here only *-∅* and *-ing* are found in the list of valid flexions. In case of homonymy (e.g., *-∅* versus *-ing*) we consider several hypotheses, which can be rejected at a further step of the algorithm.
2. If the flexion (here *-ing*) is found in the list, we apply the correspondence “flexions ⇒ sets of grammemes,” which gives us a hypothesis about the possible set of grammemes (here “verb, present participle”).
3. Then we obtain the information for the rest of the form (the potential stem, here *stopp-*) from the dictionary. This stem has been generated and added to the dictionary at the phase of the data preparation.
4. Finally, we generate the corresponding grammatical form according to our hypothesis and the dictionary information (here, the generated past participle of the verbal stem *stopp-* is *stopping*).
5. If the obtained result coincides with the input form then the hypothesis is accepted. Otherwise, the process is repeated from the step 3 with another homonymous stem (if

any) or from the step 1 with another hypothesis on the flexion.

If the grammatical form consists of several morphemes (a stem and several affixes, as described in the previous section) then the analysis process consists of several steps, precisely as generation. Again, in case of Spanish or Russian, only 3 steps are sufficient.

In the case of word form homonymy, all hypotheses are generated in the output. For example, for *writing* two hypotheses are generated: (1) a verb stem *writ-* with a verb flexion *-ing* and (2) a noun stem *writing-* with a noun flexion -∅. Further contextual disambiguation is the business of a tagger or syntactic analyzer.

As one can see, our method of analysis is not much more complex than generation. The only modules added are the model "flexions ⇒ sets of grammemes" and the module of interaction between different models.

Adding generation to the analysis algorithm does not really affect its performance since the bottleneck of any dictionary-based method is the dictionary search operation.

8 Treatment of Unknown Words

Obviously, all words with the stems *present* in the dictionary are processed correctly. The treatment of unknown words with the described architecture is also simple. We apply the same procedure of analysis to single out the hypothetical stem. If at the step 3 of the analysis algorithm described in the section 7 the stem is not found in the dictionary, we use the longest match stem (matching the letters from right to left) compatible with the given set of affixes. The longest match stem is the stem present in the dictionary that has as long as possible *ending* substring in common with the given input stem (and is compatible with the affixes already singled out).

In this way, for example, an (unknown) input string *sortifies* will be analyzed as *classifies*, i.e., as a verb, third person, present, singular—given that *classifi-* is its longest match stem for *sortifi-* (matching by *-ifi-*) compatible with the affix *-es*.

To facilitate this search, the stem dictionary is ordered by inverse order, i.e., the stems are ordered lexicographically from right to left (by the last letter, then by the next-to-last one, etc.).

Note that the systems like (Gel bukh 1992, Gelbukh 2000) based on the left-to-right order of analysis (those that first single out the stem and only then analyze the resting affixes) have to imitate this process with a special dictionary of, for example, 5-letter stem endings, since in such systems the main stem dictionary is ordered by direct order (left to right, by first letters).

No need to say that our treatment of unknown words is approximate and can suffer from both over- and undergeneration.

9 Implementation

We have implemented this methodology for Russian, which is a highly inflective language. A morphological dictionary (Zalizniak, 1980) including about 100,000 lexemes was used. Fortunately, the models for Russian morphology (rather complex) had been developed in the same dictionary. This dictionary is oriented for generation: using it, a person without any knowledge of Russian can generate any given Russian morphological form. Due to the technique we used, it proved to be possible to borrow all grammatical types from this dictionary without changing.

The implementation process took several months of work of one person. The system for Russian is available for free for academic use as Windows DLL or EXE file.

Using the same method, we are working on a morphological system for Spanish, which is also an inflective language, though not as morphologically complex as Russian. The development of the morphological model (rules, grammeme lists, etc.) has taken only several days; now we are preparing the dictionary using a semi-automatic procedure: the stems and morphological classes for the words found in the corpus are guessed automatically and in case of ambiguity the choice is made manually.

10 Conclusions

We have presented a methodology for building systems of automatic morphological analysis systems for inflective languages. The method is based on analysis through generation approach, which greatly simplifies the development. Our experience with implementation of the system for Russian shows that using our methodology the system is implemented very quickly.

The system for Russian is freely available for academic use as a Windows DLL or executable file. The Spanish version of the system is being implemented.

References

Gelbukh, A. 2002. A data structure for prefix search under access locality requirements and its application to spelling correction. To appear in *Computación y Sistemas, Revista Iberoamericana de Computación*.

Gel bukh, A.F. 1992. Effective implementation of morphology model for an inflectional natural language. J. *Automatic Documentation and Mathematical Linguistics*, Allerton Press, 26 (1): 22–31.

Hutchins, W.J., Somers, H.L. 1992. *An introduction to machine translation* , Academic Press, London.

Hausser, Ronald. 1999. Three principled methods of automatic word form recognition. Proc. of *VEXTAL: Venecia per il Tratamento Automatico delle Lingue*. Venice, Italy, Sept., pp. 91–100.

Koskenniemi, Kimmo. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki Publications, N 11.

Malkovsky, M. G. 1985. *Dialogue with an artificial intelligence system* (in Russian). Moscow State University, Moscow, Russia, 213 pp.

Sedlacek R. and P. Smrz. 2001. A new Czech morphological analyzer AJKA. *Proc. of TSD-2001*. LNCS 2166, Springer, pp. 100–107.

Sidorov, G. O. 1996. Lemmatization in automatized system for compilation of personal style dictionaries of literature writers (in Russian). *Word by Dostoyevsky* (in Russian), Moscow, Russia, Russian Academy of Sciences, pp. 266–300.

Yablonsky, S. A. 1999. Russian morphological analysis. Proc. of *VEXTAL: Venecia per il Tratamento Automatico delle Lingue*. Venice, Italy, Sept., pp 83–90.

Zalizniak, A.A. 1980. *Grammatical dictionary of Russian* (in Russian). “Russkij Jazyk”, Moscow, Russia.